Factorization in Deep Neural Networks

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Course organisation

Sessions

1. Deep Learning and Transfer Learning,
2. Quantification,
3. Pruning,
4. Factorization,
5. Distillation,
6. Operators and Architectures,
7. Embedded Software and Hardware for DL.
8. Presentations for challenge.

Course organisation

Sessions

1. Deep Learning and Transfer Learning,
2. Quantification,
3. Pruning,
4. Factorization,
5. Distillation,
6. Operators and Architectures,
7. Embedded Software and Hardware for DL.
8. Presentations for challenge.

1 Overview of unsupervised learning
- Clustering
- Decomposition using Sparse Dictionary Learning
- Decomposition using (Deep) Auto-encoders
- Manifold Learning

2 Factorization in deep neural networks

Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization. . .

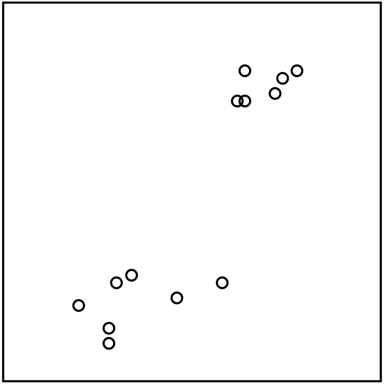
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization…



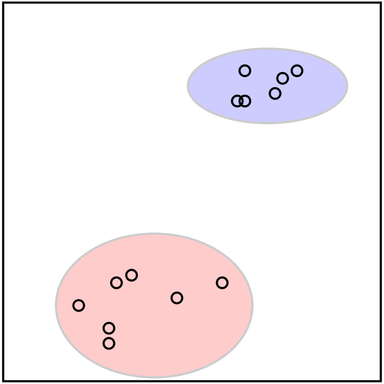
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization...



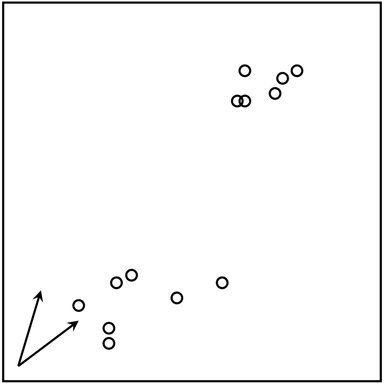
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization...



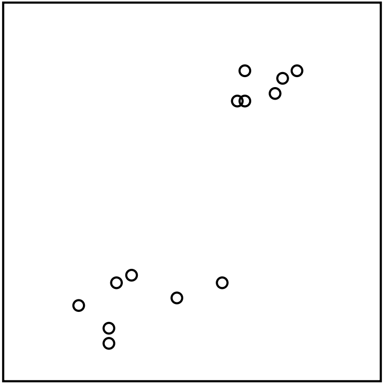
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization...



Example: clustering using L_2 norm

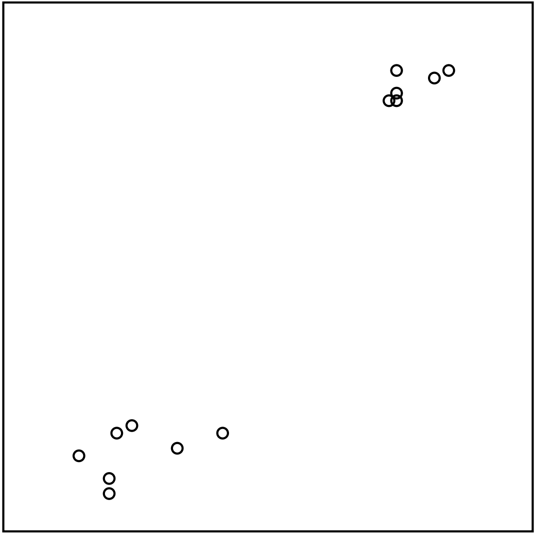
An example to perform clustering is to rely on distances to centroids. We define K *cluster centroids* $\Omega_k, \forall k \in [1..K]$

Definitions

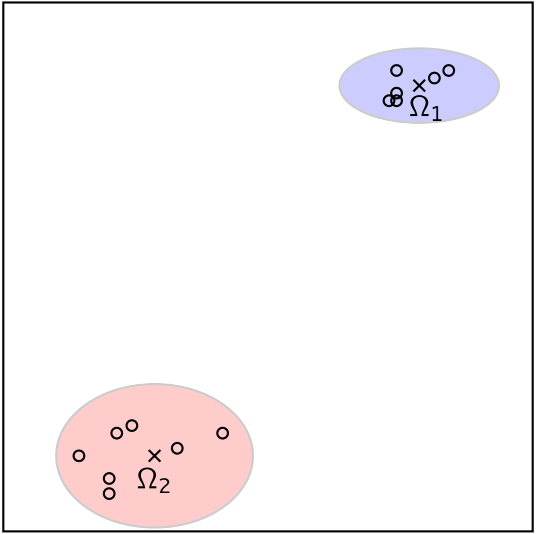
We denote $q : \mathbb{R}^d \rightarrow [1..K]$ a function that associates a vector $\mathbf{x}$ with the index of (one of) its closest centroid $q(\mathbf{x})$. Formally:

- $\forall k \in [1..K], \Omega_k \in \mathbb{R}^d$
- $\forall \mathbf{x} \in X, \forall j \in [1..K], \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2 \leq \|\mathbf{x} - \Omega_j\|_2$
- Error $E(q) \triangleq \sum_{\mathbf{x} \in X} \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2$
- $X = \bigcup_k \underbrace{\{\mathbf{x} \in X, q(\mathbf{x}) = k\}}_{\text{cluster } k}$

Example: clustering using L_2 norm



Example: clustering using L_2 norm



Clustering using L_2 norm

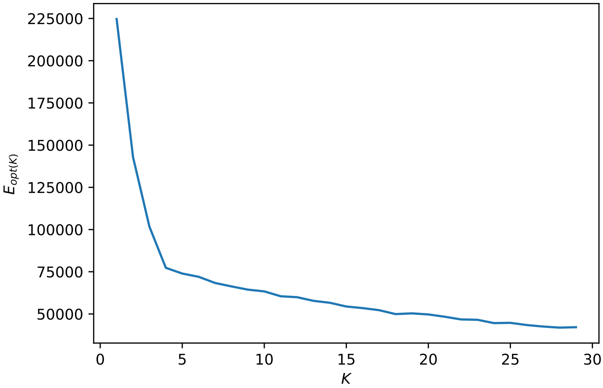
Quantizing MNIST

- Replace $\mathbf{x}$ by $\Omega_{k(\mathbf{x})}$
- Compression factor $\kappa = 1 - K/N$

Clustering using L_2 norm

Choosing K

- Finding a compromise between error and compression,
- Simple practical method : "elbow".

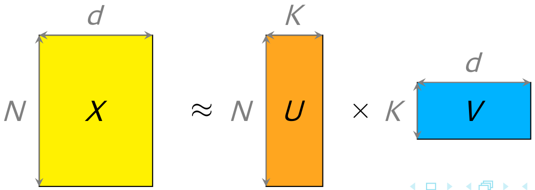


Sparse Dictionary Learning

Definitions

Dictionary learning solves the following matrix factorization problem:

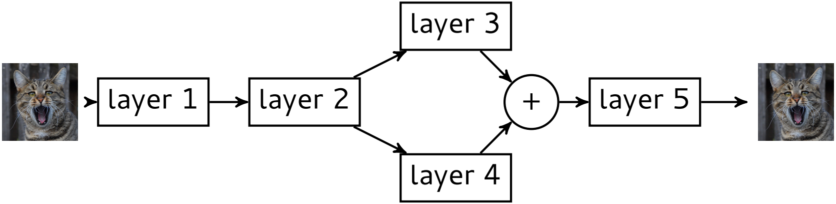
- The set X is considered as a matrix $X \in \mathcal{M}_{N \times d}(\mathbb{R})$,
- We consider decompositions using a dictionary $V \in \mathcal{M}_{K \times d}(\mathbb{R})$ and a code $U \in \mathcal{M}_{N \times k}(\mathbb{R})$, with the lines of V being with norm 1,
- Error $E(U, V) \triangleq \|X - UV\|_2 + \alpha\|U\|_1$
- Training: find U^*, V^* that minimizes $E(U^*, V^*)$
- α is a sparsity control parameter that enforces codes with soft (ℓ_1) sparsity



(Deep) auto-encoders

Inputs/outputs

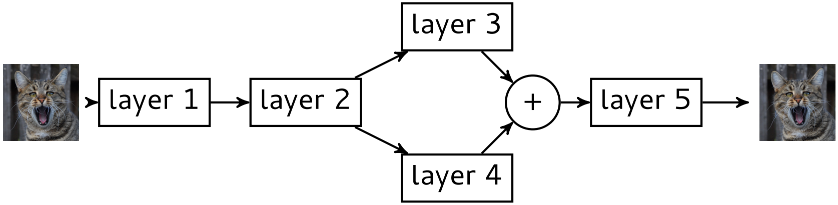
- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



(Deep) auto-encoders

Inputs/outputs

- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



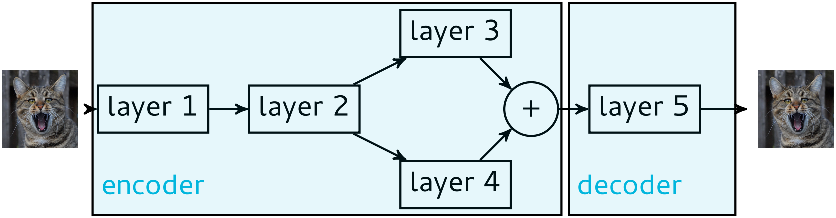
Precisions

- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error**: $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

(Deep) auto-encoders

Inputs/outputs

- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



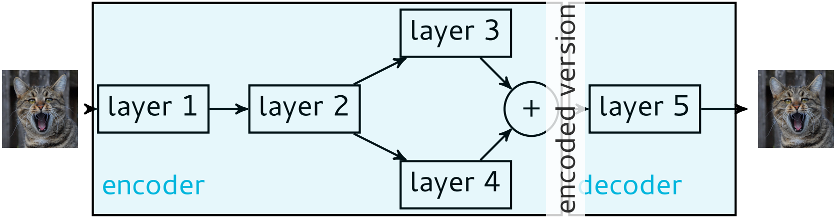
Precisions

- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error**: $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

(Deep) auto-encoders

Inputs/outputs

- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



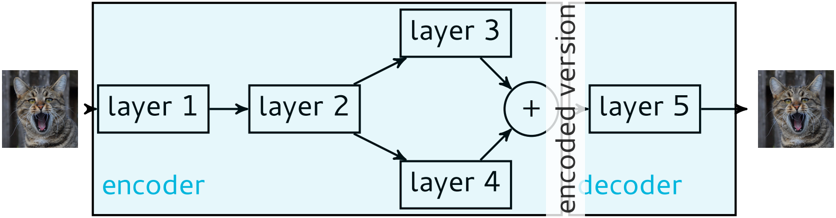
Precisions

- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error**: $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

(Deep) auto-encoders

Inputs/outputs

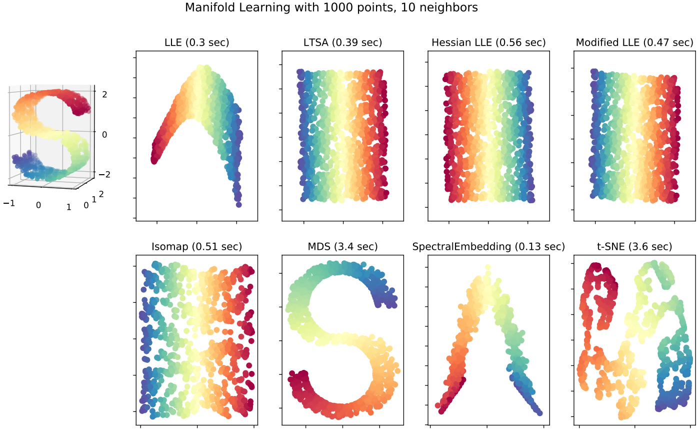
- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



Precisions

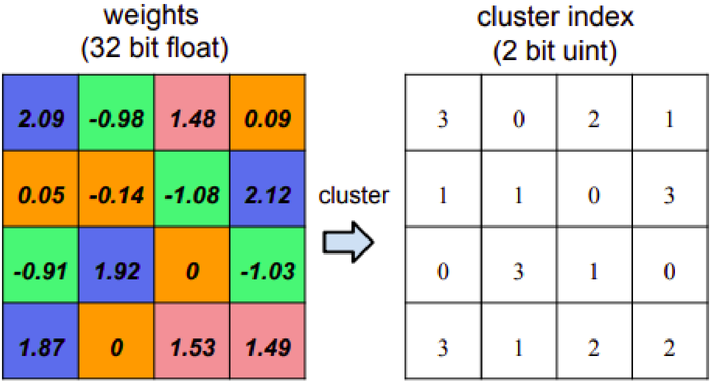
- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error**: $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

Manifold Learning



Approaches to uncover lower dimensional structure of high dimensional data. Source : Manifold module, sklearn website

Using clustering to factorize a network



from `https://arxiv.org/abs/1510.00149`

Pruning and compressing neural networks while training

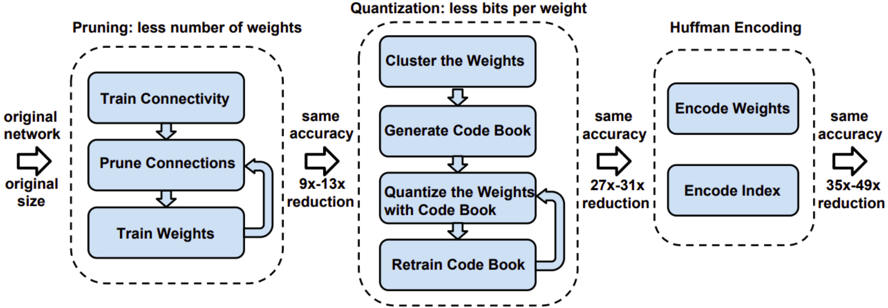


Figure 1: The three stage compression pipeline: pruning, quantization and Huffman coding. Pruning reduces the number of weights by $10\times$, while quantization further improves the compression rate: between $27\times$ and $31\times$. Huffman coding gives more compression: between $35\times$ and $49\times$. The compression rate already included the meta-data for sparse representation. The compression scheme doesn't incur any accuracy loss.

Results only on LeNet and VGG...

`https://arxiv.org/abs/1510.00149`

Deep k-means

Principle

- Training
- Row-wise k-means clustering for parameters (per layer)
- Re-training using k-means regularization

Model	Δ (%)	CR
Soft Weight-Sharing	-2.02	45
Deep k-Means WR	-16.02	45
Deep k-Means WR	-25.45	47
Deep k-Means WR	-45.08	50
Deep k-Means	-1.63	45
Deep k-Means	-2.23	47
Deep k-Means	-4.49	50

Table 3. Compressing Wide ResNet in comparison to soft weight-sharing (Ullrich et al., 2017).

https://arxiv.org/abs/1806.09228
Similar to Ulrich et al. 2017 (https://arxiv.org/abs/1702.04008) which used soft-weight sharing using Gaussian Mixture Models.